



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Gº en Ingeniería en Informática



Trabajo final del Gº Ing. Informática:
Plataforma para la realización de torneos
2x2



Presentado por Jose Antonio Barbero Aparicio
en enero de 2017
Tutor José Ignacio Santos Martín y Virginia Ahedo
García



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Gº en Ingeniería en Informática



D. José Ignacio Santos Martín y Virginia Ahedo García,
profesores del departamento de Ingeniería Civil, área de
Organización y Gestión de Empresas

Exponen:

Que el alumno D. Jose Antonio Barbero Aparicio, con
DNI 71301203F, ha realizado el Trabajo final del
GºIng.Informática titulado: Plataforma para la
realización de torneos 2x2.

y que dicho trabajo ha sido realizado por el alumno bajo
la dirección de los que suscriben, en virtud de lo cual, se
autoriza su presentación y defensa.

En Burgos a 10 de enero de 2017

José Ignacio Santos Martín y Virginia Ahedo García



Índice de contenido

Índice de ilustraciones.....	1	2.5.HTML.....	18
Índice de tablas.....	1	2.6.JavaScript.....	18
I -Introducción.....	4	2.6.A.jQuery.....	18
II -Objetivos del proyecto.....	5	2.6.B.Ajax.....	18
1.Objetivos funcionales.....	5	2.7.PHP.....	18
2.Objetivos técnicos.....	5	2.8.CSS.....	19
3.Objetivos académicos.....	5	2.9.Bootstrap.....	19
III -Conceptos teóricos.....	7	2.10.SublimeText.....	19
1.Teoría de juegos.....	7	2.10.A.Plugin Emmet.....	20
1.1.Tipos de juegos.....	8	2.10.B.Alternativa1: Brackets.....	20
1.1.A.Formas de representación.....	8	2.10.C.Alternativa2: Atom.....	20
1.1.B.Cooperativos.....	8	2.10.D.Alternativa3: Eclipse.....	20
1.1.C.Simétricos o asimétricos.....	8	2.11.Mendeley.....	20
1.1.D.Suma cero o suma no cero.....	9	2.11.A.Alternativa: Zotero.....	20
1.1.E.Iterados.....	9	2.12.LibreOffice Writer.....	21
1.2.Dilema del prisionero.....	9	2.12.A.Alternativa1: OpenOffice	21
1.3.Equilibrio de Nash.....	10	Writer.....	21
1.4.Torneo de Axelrod.....	12	2.12.B.Alternativa2: Microsoft Word	21
IV -Técnicas y herramientas.....	15	21
1.Metodología utilizada.....	15	2.12.C.Alternativa3: LaTeX.....	21
2.Herramientas Software.....	16	V -Aspectos relevantes del desarrollo del pro-	22
2.1.GitHub.....	16	yecto.....	22
2.1.A.Alternativa1: Bitbucket.....	16	1.Modelo individual.....	22
2.1.B.Alternativa2: Sourceforge.....	16	2.Modelo global.....	23
2.1.C.Alternativa3: GitLab.....	16	3.Aplicación web.....	24
2.2.ZenHub.....	16	3.1.Servidor web.....	25
2.2.A.Alternativa1: Trello.....	16	VI -Trabajos relacionados.....	26
2.2.B.Alternativa2: Sprintometer.....	16	VII -Conclusiones y líneas de trabajo futuras	28
2.3.NetLogo.....	17	28
2.3.A.NetLogo Behaviour Space.....	17	1.Conclusiones.....	28
2.3.B.NetLogo Web.....	17	2.Líneas de trabajo futuras.....	28
2.4.XAMPP.....	17	VIII -referencias.....	29
2.4.A.Alternativa1: MAMP.....	18	Bibliografía.....	29
2.4.B.Alternativa2: WAMP.....	18		

Índice de ilustraciones

Illustration 1: Representación de un juego en forma extensiva[18].....	8
Illustration 2: Solución ante la falta de soporte de runresult por parte de NetLogo web.....	23

Illustration 3: Solución al problema de compatibilidad del modelo NetLogo web con html5.....	23
--	----

Índice de tablas

Table 1: Matriz de pagos (representación en forma estratégica) del juego “piedra, papel o tijeras”[11].....	8
Table 2: Matriz de recompensas de un juego simétrico 2 x 2.....	9
Table 3: Matriz de pagos del dilema del prisionero.....	10

Table 4: Resultados de la primera ronda del torneo celebrado por Axelrod. La explicación en profundidad de cada una de las estrategias puede encontrarse en el artículo original.[14]	12
---	----







Resumen

Se desea desarrollar una herramienta docente de tipo web que proporcione a un profesor y a sus alumnos un medio en el que simular torneos 2 x 2. El objetivo principal es ilustrar el aprendizaje de la teoría de juegos en cualquiera de las múltiples disciplinas influenciadas por esta rama de conocimiento.

Para llevar a cabo este propósito se ha efectuado una combinación entre dos tipos diferentes de tecnologías. Las primeras, de tipo web, están dirigidas al desarrollo de la propia plataforma sobre la que se sustenta la aplicación. El segundo tipo, enfocado a la simulación, se centra en la propia reproducción del torneo.

Debido a que se trata de una herramienta destinada al aprendizaje, se ha tenido en cuenta el proceso de adaptación de los alumnos a la metodología del torneo. De esta manera, se incluye un simulador de mayor simpleza con el que se busca permitir a los alumnos un proceso de prueba de estrategias. Se establece, así, un enfrentamiento contra soluciones predefinidas antes de pasar al torneo contra otros participantes.

Finalmente, se ha creado un modelo en el que realizar la simulación del torneo final. Este torneo interpreta las estrategias proporcionadas por los participantes y, a partir de ellas, realiza una serie de emparejamientos. El objetivo es que todas las estrategias se enfrenten entre sí una sola vez. Para concluir, recurriendo a las puntuaciones registradas en cada emparejamiento, se genera un conjunto de puntuaciones globales, a partir de ellas, se establecen las estrategias más óptimas dentro de ese entorno concreto.

Abstract

We want to develop a web-based educational tool in order to bring to the teacher and the students a mean capable of simulate 2 x 2 tournaments. The main goal is to support the teaching of game theory in any of the several fields influenced by this branch of knowledge.

In order to carry out this purpose we have combined two different technology types. The first one, web-based, is focused to develop the web platform in which the application is standing. The second type, based on simulation, is focused on the tournament execution.

Since it is a learning tool, we have taken into account the adaptation process of the students to the tournament. In this way, a simpler simulator has been developed in order to bring the student a way to test his strategies. So a confrontation against predefined solutions is established before the tournament with solutions of the rest of participants.

Finally a model has been created to perform the final tournament simulation. This tournament interprets the strategies provided by the participants and performs pairings in order to confront the strategies with each other one. To conclude, the tournament uses the scores recorded in every single pairing and creates a set of global scores. Based on those scores the most optimal strategies in that particular environment are established.





I - INTRODUCCIÓN

Las nuevas tecnologías están llevando a la docencia a incorporar ciertas metodologías basadas en herramientas digitales. La utilización de experimentos en forma de juegos en el aula ha demostrado ya buenos resultados. El paso siguiente, los simuladores y juegos ejecutados en computadores, están dando resultados del mismo tipo.

A su vez, la teoría de juegos es un campo que sigue creciendo en la actualidad. Además, la naturaleza de sus conceptos encaja a la perfección con una forma de enseñanza basada en simuladores o juegos.

Del mismo modo, vemos que la enseñanza online está experimentando un crecimiento en expansión gracias, tanto a las universidades con enseñanza a distancia, como a los portales web basados en cursos, como Coursera o Miriada X.

Por estas razones se ha planteado el desarrollo de una herramienta docente de simulación virtual online destinada a la explicación de la teoría de juegos a los alumnos. Se centra en el planteamiento de juegos de tipo 2 x 2, el tipo de juego del conocido dilema del prisionero, un caso ampliamente utilizado para ejemplificar la teoría de juegos. El objetivo es que un profesor pueda crear torneos en los que puedan participar sus alumnos desde sus ordenadores. El profesor tiene libertad para fijar los parámetros del torneo. Tanto los alumnos como el profesor podrán consultar sus puntuaciones al final del torneo para sacar sus conclusiones. El objetivo que se fija a los alumnos es que desarrollen su propia estrategia en forma de código con la que obtener mayor puntuación que el resto de participantes. Se les proporcionará un medio con el que probar esa estrategia contra algunas ya preestablecidas.

Con este proyecto se desea dotar a los profesores y a los alumnos de una manera de afrontar la teoría de juegos desde las aulas más amena e ilustrativa para los estudiantes, y con más oportunidades para los docentes.



II - OBJETIVOS DEL PROYECTO

1. *Objetivos funcionales*

Los principales objetivos relacionados con la funcionalidad de la aplicación desarrollada en el proyecto son los siguientes:

- Dotar a alumnos y docentes de una herramienta capaz de complementar la explicación de los conceptos asociados a la teoría de juegos, mediante la posibilidad de celebrar torneos de tipo 2 x 2 en una plataforma web.
- Permitir al profesor definir el tipo de juego 2 x 2 estableciendo los parámetros de recompensa del juego (payoff).
- Mostrar los resultados de un torneo tras su conclusión.
- Almacenar de los resultados en una base de datos al finalizar cada torneo.
- Permitir a los participantes del torneo probar sus soluciones contra una serie de estrategias estándar preestablecidas antes de enviar su respuesta.
- Permitir a los profesores registrarse como administrador para poder crear y finalizar torneos.
- Permitir al creador del torneo poder finalizarlo.
- Permitir a los alumnos registrarse como participante para poder formar parte de los torneos y consultar sus participaciones en torneos pasados.
- Dar la posibilidad de que existan varios torneos abiertos al mismo tiempo.

2. *Objetivos técnicos*

Los objetivos principales de aspecto técnico que se desean satisfacer a lo largo del proyecto son los siguientes:

- Funcionamiento adecuado en los principales navegadores. Esto incluye, en principio, Firefox, Safari y Chrome.
- Ejecución de la aplicación sobre un servidor local creado con XAMPP.
- Evitar la instalación de NetLogo en el ordenador particular de los usuarios gracias a la utilización de NetLogo Web.
- Diseño responsive de la aplicación mediante el uso de Bootstrap.
- Ejecución del torneo en un tiempo aceptable, a pesar de que la carga en el momento de la puesta en común de todas las estrategias conlleve un coste relativamente alto.

3. *Objetivos académicos*



El proceso de desarrollo del proyecto, además de buscar completar objetivos técnicos y de



funcionalidad, tiene una parte orientada al propio aprendizaje. Los objetivos relacionados con esa parte del proyecto son los siguientes:

- Profundizar en el conocimiento de la teoría de juegos, así como en los autores que han aportado avances en ella a través de artículos o libros.
- Aprender a crear un servidor local desde cero en el que se sustente la aplicación completa.
- Profundizar en los conceptos aprendidos sobre desarrollo web, a través de la creación de una aplicación web completa. En especial en el uso de tecnologías que antes no se habían utilizado de forma considerable, como JavaScript, PHP o MySQL.
- Ahondar en la utilización de tecnologías que pueden orientarse a investigación. En este caso, NetLogo.
- Aplicar todo lo estudiado sobre repositorios de código y control de versiones a través del uso de estas herramientas en un proyecto completo.
- Poner en práctica una metodología ágil en un desarrollo real de tamaño considerable.



III - CONCEPTOS TEÓRICOS

Con el objetivo de entender más claramente el fondo del proyecto, se van a detallar a continuación los conceptos más relevantes, alrededor de los cuales gira el contenido del mismo.

1. Teoría de juegos

“La teoría de juegos estudia situaciones de conflicto y cooperación a las que denominamos juegos, en las que interactúan individuos racionales, analizando los comportamientos y resultados que son de esperar, bien mediante decisiones individuales (caso de los juegos no cooperativos), bien mediante acuerdos entre los participantes (caso de los juegos cooperativos).” [2]

A continuación vamos a estudiar un breve repaso [3] al proceso de desarrollo de conocimiento de la teoría de juegos para ayudar a contextualizar el proyecto. A pesar de que existen algunos conceptos relacionados con la teoría de juegos con un origen anterior, se considera que las bases de la teoría de juegos fueron establecidas por el economista Oskar Morgenstern y el matemático John von Neumann a través de la publicación de *Theory of Games and Economic Behaviour* (1944) [4]. Con ella se planteaban los conceptos de juegos de suma cero y de juegos cooperativos. En 1950 Melvin Dresher y Merrill Flood plantean el experimento del famoso dilema del prisionero, uno de los ejemplos más utilizados para ilustrar la teoría de juegos en juegos cooperativos de suma no nula. Entre los mayores contribuidores en el campo de la teoría de juegos, destaca John F. Nash. Nash hizo varias de sus colaboraciones más importantes entre 1950 y 1953, con la publicación de cuatro artículos [5]–[8] en los que introducía por primera vez el equilibrio que más tarde llevaría su nombre, además de otras numerosas aportaciones en los juegos de tipo no cooperativo. Fue posteriormente, en 1994, cuando comenzó a llegar el reconocimiento a través de John C. Harsanyi, Reinhard Selten y el propio John Nash, quienes fueron galardonados con el premio Nobel de Economía, acarreando un auge para este campo de estudio.

El hecho de que la teoría de juegos es un campo de conocimiento en expansión queda demostrado observando que las publicaciones académicas relacionadas con este campo han ido en aumento. Una simple búsqueda en Google Scholar del término “Game Theory” devuelve casi tres millones de resultados [9].

A pesar de que la teoría de juegos surgió inicialmente en el campo de la economía, ha acabado por establecer corrientes de investigación en campos como la informática, la psicología, la política o la biología. Es esta última la que más se ha visto influenciada, en palabras de John Maynard Smith: “Paradójicamente, ha resultado que la teoría de juegos se aplica más fácilmente a la biología que al campo de la economía del comportamiento, para el que fue originalmente diseñada. Hay dos razones para ello. La primera, la teoría requiere que los valores de los diferentes resultados [...] puedan ser medidos en algún tipo de escala. En el caso de los humanos, esta medida viene dada por la ‘utilidad’ - un concepto artificial e incomodo: en biología, la adaptación darwiniana proporciona una medida natural y unidimensional. En segundo lugar, y más importante, en la búsqueda de la solución de un juego, el concepto de la racionalidad humana es sustituido por la estabilidad evolutiva. La ventaja de esto es que hay buenas razones teóricas para esperar que las poblaciones evolucionen en estados estables, mientras que hay suficientes motivos para dudar de que los humanos se comporten siempre racionalmente” [10].





1.1. Tipos de juegos

1.1.A. Formas de representación

Los juegos pueden representarse de dos modos [2]:

- En forma extensiva: se trata de una representación en forma de árbol. Se utilizan una serie de nodos (situaciones de elección), acciones (enlaces entre nodos) y vectores de pagos (con dos datos que definen lo que recibe cada jugador).

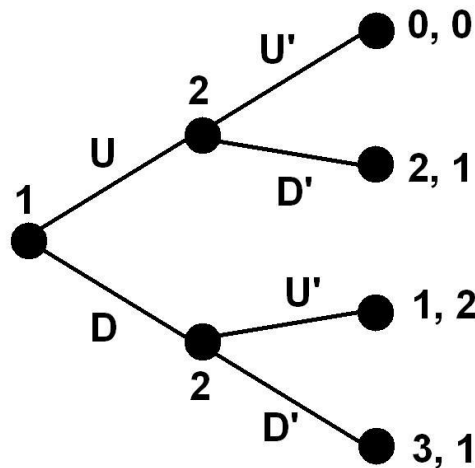


Illustration 1: Representación de un juego en forma extensiva[18]

- En forma estratégica: el juego se define de tal forma que una estrategia de cierto jugador hace corresponder a cada conjunto de información del propio jugador, con una de las acciones especificadas en el conjunto de información dado [2]. La información se suele plantear en una tabla similar a la siguiente, denominada matriz de pagos.

	Piedra	Papel	Tijeras
Piedra	(0, 0)	(-1, 1)	(1, -1)
Papel	(1, -1)	(0, 0)	(-1, 1)
Tijeras	(-1, 1)	(1, -1)	(0, 0)

Table 1: Matriz de pagos (representación en forma estratégica) del juego “piedra, papel o tijeras” [11].

1.1.B. Cooperativos

En la publicación de *Theory of Games and Economic Behaviour* [4] se establece la diferenciación entre juegos cooperativos y no cooperativos. Se trata de juegos en los que se ofrece la posibilidad de que los jugadores lleguen a un acuerdo (el cual quedarían obligados a cumplir), a raíz del que ambos salen beneficiados [2].

1.1.C. Simétricos o asimétricos

“Decimos que un juego es simétrico en el caso de que el payoff de un jugador se pueda expresar como una transposición del payoff del otro jugador. Si la transposición de la matriz del otro



jugador es ordinalmente simétrico, entonces se trata de un juego ordinalmente simétrico” [12].

La matriz de recompensas de un juego simétrico seguiría el esquema siguiente:

	1	2
1	(a, a)	(b, c)
2	(c, b)	(d, d)

Table 2: Matriz de recompensas de un juego simétrico 2 x 2

1.1.D. Suma cero o suma no cero

Los juegos de suma cero se caracterizan por un equilibrio entre los beneficios de un jugador y las pérdidas del otro. “Modelizan situaciones de conflicto puro entre dos jugadores, en las cuales lo que un jugador gana es exactamente lo que su contrincante pierde” [2].

La matriz de recompensas de un juego de suma cero seguiría la estructura de la tabla siguiente:

	1	2
1	(a, -a)	(-b, b)
2	(-c, c)	(d, -d)

1.1.E. Iterados

A raíz de las limitaciones de los tipos de juegos estudiados, se terminó introduciendo un tipo de modificación que consiste en aumentar el número de enfrentamientos de los torneos de manera ilimitada. En el caso de los torneos que se realizan en este proyecto, el número de rondas del torneo es determinado en el momento de su creación.

1.2. Dilema del prisionero

El dilema del prisionero es uno de los ejemplos más utilizados para ilustrar la explicación de la teoría de juegos. Se trata de un juego cooperativo de dos jugadores, de suma no nula. Consiste en que dos personas han sido retenidas por cierto delito. A pesar de ello, no se tienen pruebas suficientes contra ellas para condenarlas por lo que han hecho. A partir de esta situación a los prisioneros se les ofrece la posibilidad de delatar¹ al otro prisionero (para así reducir su propia condena y aumentar la de su contrincante) o de cooperar con el otro jugador (para así intentar reducir la pena de ambos). En el caso de que ambos se delaten mutuamente, se les asignará a ambos una pena mayor que la que tendrían si el otro no les hubiese delatado, pero menor que si ellos no hubiesen confesado, y su rival sí lo hubiese hecho. Para ver más claramente las posibles decisiones y sus recompensas, veamos la matriz de pagos del dilema.





Jugador 1 \ Jugador 2	Cooperar	Delatar
Cooperar	(3, 3)	(0, 5)
Delatar	(5, 0)	(1, 1)

Table 3: Matriz de pagos del dilema del prisionero²

Robert Axelrod [1] estableció un serie de nombres a cada una de las recompensas para ayudar a su comprensión:

- “Reward for mutual cooperation”: es la recompensa recibida por ambos jugadores cuando los dos cooperan entre sí. Es la forma que permite maximizar el beneficio de ambos a la vez.
- “Temptation to defect”: es la recompensa obtenida por el jugador que delata a su rival, en el caso de que este último haya decidido cooperar. Es la forma que maximiza los beneficios de un solo jugador.
- “Sucker’s payoff”: es la recompensa que recibe un jugador cuando decide cooperar con su rival, pero este último le ha delatado. Es la peor recompensa posible para un jugador.
- “Punishment for mutual defection”: es la recompensa dada a ambos jugadores cuando ninguno de ellos coopera y ambos se delatan mutuamente. Se trata de una recompensa baja para ambos, pero esta situación sería la que se establecería por un equilibrio de Nash en este dilema.

Robert Axelrod realiza en esta cita un resumen perfecto de la forma individual de razonar de los jugadores, dejando perfectamente claro, por qué se le llama dilema:

“Es mejor para ti delatar si tú crees que el otro jugador va a cooperar. Y también es mejor para ti delatar si crees que el otro jugador va a delatarte. En cualquier caso, para ti, es mejor delatar.

Hasta aquí, bien. Pero la misma lógica puede aplicarse también a tu rival. [...] Por lo tanto, los dos deberíais delataros. Sin embargo obtendréis un punto, que es peor que los tres puntos que obtendríais si ambos cooperaseis. La racionalidad individual lleva a un resultado subóptimo para ambos. He aquí el dilema.” [1]

1.3. Equilibrio de Nash

El equilibrio de Nash en estrategias puras definido formalmente, sigue la descripción siguiente:

“En un juego $G = \{S_1, \dots, S_n; u_1, \dots, u_n\}$, decimos que el perfil de estrategias puras $(s_1^*, s_2^*, \dots, s_i^*, \dots, s_n^*)$ es un Equilibrio de Nash si para cada jugador i , $u_i(s_1^*, \dots, s_{i-1}^*, s_i^*, s_{i+1}^*, \dots, s_n^*) \geq u_i(s_1^*, \dots, s_{i-1}^*, s_i, s_{i+1}^*, \dots, s_n^*)$ para todo s_i de S_i .

² Se han utilizado los mismos pagos establecidos en *The Evolution of Cooperation* de Robert Axelrod.[1]



Es decir, para cada jugador i , s_i^* es una solución del problema $\max u_i(s_1^*, \dots, s_{i-1}^*, s_i, s_{i+1}^*, \dots, s_n^*)$ donde s_i es la variable de decisión y pertenece a S_i . O dicho de otro modo, para cada jugador i , s_i^* es una respuesta óptima a s_{-i}^* . [2]

Si nos atenemos a una definición algo más informal, un Equilibrio de Nash sería lo siguiente:

Un equilibrio de Nash es un concepto de teoría de juegos en el que el resultado de un juego es aquel en el que ninguno de los jugadores, teniendo en cuenta la estrategia que lleva a cabo su rival, está motivado a desviarse de la estrategia que ha elegido. Cualquiera de los jugadores no recibirá beneficios en el caso de que modificase su estrategia y su rival continuase con la elección que ya había tomado. Un juego puede tener varios equilibrios de Nash, o no tener ninguno. [13]

Que un juego termine por llegar al equilibrio de Nash no tiene por qué ser la solución que más beneficia a ambos jugadores, sino que cada jugador tiene la estrategia más óptima considerando las soluciones que están llevando a cabo el resto de individuos.

Para ilustrar la idea de equilibrio de Nash, veamos un ejemplo con el dilema del prisionero que acabamos de tratar. En el caso del dilema del prisionero, la intuición nos dice que el equilibrio se encuentra en el punto en el que ambos cooperan, de tal forma que ambos maximizan sus beneficios. Sin embargo, esto es un error. En el caso de que uno de los jugadores decidiese traicionar al otro, obtendría un beneficio. Por lo tanto, sí que existen incentivos para que los jugadores quieran cambiar su estrategia en esa situación. El equilibrio de Nash correcto en el caso del dilema del prisionero es que ambos jugadores delaten al otro. De esta forma ninguno de los dos obtiene el máximo beneficio, pero ninguno tiene motivación para cambiar de estrategia, ya que si el otro jugador sigue con la solución que había adoptado hasta ahora, el jugador que realiza el cambio saldría penalizado.





1.4. Torneo de Axelrod

Ranking	Nombre del participante	Nombre de la estrategia	Campo de estudio	Longitud de la solución (líneas de código)	Puntuación final
1	Anatol Rapoport	TIT FOR TAT	Psicología	4	504.5
2	Nicholas Tideman y Paula Chieruzzi	TIDEMAN AND CHIERUZZI	Economía	41	500.4
3	Rudy Nydegger	NYDEGGER	Psicología	23	485.5
4	Bernard Grofman	GROFMAN	Ciencias Políticas	8	481.9
5	Martin Shubik	SHUBIK	Economía	16	480.7
6	William Stein y Amnon Rapoport	STEIN AND RAPOPORT	Matemáticas y Psicología	50	477.8
7	James W. Friedman	FRIEDMAN	Economía	13	473.4
8	Morton Davis	DAVIS	Matemáticas	6	471.8
9	James Graaskamp	GRAASKAMP		63	400.7
10	Leslie Downing	DOWNING	Psicología	33	390.6
11	Scott Feld	FELD	Sociología	6	327.6
12	Johann Joss	JOSS	Matemáticas	5	304.4
13	Gordon Tullock	TULLOCK	Economía	18	300.5
14	Anónimo		Ciencias Políticas	77	282.2
15		Random		5	276.3

Table 4: Resultados de la primera ronda del torneo celebrado por Axelrod. La explicación en profundidad de cada una de las estrategias puede encontrarse en el artículo original. [14]

En 1980 Robert Axelrod reunió a numerosos expertos de varias áreas del conocimiento con el objetivo de enfrentarlos en un torneo. Este torneo tenía como base el dilema del prisionero iterado. En él, los participantes debían desarrollar una estrategia que, partiendo de unos criterios iniciales, pueda elegir un movimiento dependiendo de las elecciones que haya tomado su rival. La estrategia queda establecida desde el inicio y no puede modificarse en torno a las elecciones que tome el contrincante.

El objetivo era la obtención de la máxima puntuación final posible después de enfrentarse a cada una de las otras estrategias por separado, a sí misma y a una estrategia de toma de decisiones aleatorias, durante exactamente doscientos movimientos en cada enfrentamiento.

El ganador del torneo fue Anatol Rapoport, con una estrategia llamada “Tit for Tat” realmente simple (4 líneas de código) y con un rendimiento que promediaba 504 puntos por ronda (200 movimientos), siendo el máximo 1000 y el mínimo 0. Por lo tanto, no es la complejidad de una estrategia lo que la hace más eficiente sino cuatro características fundamentales:



1. **Amabilidad:** son amables aquellas estrategias que nunca son las primeras en delatar. Se trata de una característica tan importante para un buen rendimiento, que las ocho primeras estrategias en la clasificación del torneo eran amables. Ninguna de las demás lo era. En cuanto a puntuaciones, las estrategias amables promediaban entre 472 y 504 puntos mientras que las que no eran amables promediaban 401 puntos. Otro asunto importante es que cuando dos estrategias amables se cruzan entre sí, ambas terminan obteniendo una puntuación de 600 puntos, ya que ninguna delata a su rival, maximizando así la puntuación obtenida por ambas.
2. **Indulgencia:** es una propiedad que define en qué medida una estrategia es propensa a cooperar cuando el rival la ha delatado en el anterior movimiento. La estrategia amable que peor puntuación obtuvo fue aquella que menos indulgencia mostraba. La razón para que las estrategias que no eran amables obtuviesen peor puntuación viene dada porque la mayoría de las estrategias presentadas al torneo eran muy poco indulgentes.
3. **Deseo de venganza:** define la velocidad con la que una estrategia delata a la otra a raíz de una traición previa inesperada. Es una forma de definir cómo una estrategia responde a una provocación por parte de otra. Sirve como un tipo de diferenciación dentro de las estrategias amables.
4. **Claridad:** establece la facilidad con la que las demás estrategias pueden reconocer la forma de actuar de cierta estrategia dada.

A continuación vamos a ver algunas de las estrategias que podrían participar en un torneo de este tipo:

- **Siempre cooperar:** una estrategia simple, el jugador que la utiliza coopera con su oponente bajo cualquier circunstancia.
- **Siempre delatar:** otra estrategia simple, el jugador siempre delata a su rival, tome este las decisiones que tome.
- **Tit for Tat:** esta estrategia consiste en colaborar en el primer movimiento y después repetir el movimiento que haya realizado el oponente en la jugada anterior. Es la estrategia que resultó vencedora en ambas rondas del torneo de Axelrod. Cabe destacar que era la estrategia con menos líneas de código presentada al torneo. El éxito de Tit for Tat reside en que es una estrategia amable, indulgente, vengativa y clara al mismo tiempo.
- **Tit for Two Tats:** es una variante de Tit for Tat que habría ganado el torneo si se hubiera presentado en primera ronda, sin embargo fue presentado en segunda ronda por John Maynard Smith y no obtuvo un buen resultado en ese entorno. Se trata de una opción algo más indulgente que mejora la puntuación de Tit for Tat en el entorno de la primera ronda del torneo de Axelrod. En este caso, en lugar de hacer la jugada que hizo el oponente en el turno anterior, le da un movimiento más de margen. De esta forma, decide tomar la decisión que tomó su contrincante dos rondas antes. Esto la convierte en una estrategia más indulgente que el Tit for Tat original.
- **Friedman:** es la estrategia amable con peor rendimiento de la primera ronda. Esto se debe a que es el perfecto ejemplo de estrategia poco indulgente. Inicialmente Friedman coopera, pero en el momento en el que su rival le delata, deja de cooperar hasta el final de la partida.
- **Joss:** es una variante de Tit for Tat que intenta sacar beneficio de los momentos en los que su rival se ha decidido a cooperar. Empieza cooperando, y cuando su rival le delata, Joss repite ese movimiento. Sin embargo, cuando su oponente coopera, en lugar de imitar el movimiento cooperativo, existe un 10% de probabilidades de que le delate en lugar de





cooperar.

- **Downing:** es una regla basada en analizar el comportamiento de su rival, obteniendo así información acerca de qué movimiento va a realizar.
- **Tester:** es una regla orientada a beneficiarse de las estrategias más débiles ante ello, sin embargo recula si encuentra una estrategia que no va a poder explotar. Inicialmente delata para comprobar el comportamiento de su oponente. Si este coopera, continua delatando. En el caso de que el otro jugador le delate, continua el resto de rondas comportándose como si fuese Tit for Tat.
- **Tranquilizer:** también se orienta a obtener beneficio de aquellas estrategias que lo permiten. Inicialmente busca obtener una situación estable en la que ambos jugadores cooperan. Cuando ha conseguido llegar a ese estado durante un número suficiente de rondas, intenta obtener beneficio delatando a su rival una ronda. Mientras esté obteniendo una puntuación de 2.25 puntos por ronda, no delatará dos veces seguidas ni delatará más de una cuarta parte de las rondas.
- **Random:** es una estrategia con el 50% de probabilidad de cooperar. Se introdujo en el torneo independientemente de que alguien la presentase.

Es importante tener en cuenta que el rendimiento de todas y cada una de las estrategias depende de las demás, del entorno en el que se desarrolla el torneo. Aunque Tit for Tat ganase el torneo, si se situase a esta estrategia en un entorno rodeado de estrategias aleatorias, Tit for Tat obtendría una puntuación mucho más baja. Esto se debe a que sería explotada por los cambios de decisión arbitrarios de la estrategia aleatoria. La ventaja de Tit for Tat no viene de obtener más puntuación que sus rivales. Tit for Tat nunca obtiene mayor puntuación en un enfrentamiento directo. Su éxito se basa en obtener más puntuación global que otras estrategias que no intentan maximizar el beneficio con sus rivales. [1], [14]–[17]



IV - TÉCNICAS Y HERRAMIENTAS

1. Metodología utilizada

A lo largo del desarrollo del proyecto se ha seguido una metodología ágil de trabajo. Este tipo de metodologías se identifican por seguir un proceso de desarrollo de tipo iterativo e incremental. Están representadas por cuatro principios [1] que establecen su filosofía:

1. Los individuos y su interacción por encima de procesos y herramientas.
2. El software que funciona frente a la documentación exhaustiva.
3. La colaboración con el cliente por encima de la negociación contractual.
4. La respuesta al cambio por encima del seguimiento de un plan.

En concreto, se ha seguido la metodología Scrum [2], representada por una división del trabajo en forma de sprints: periodos de tiempo al final de los cuales, se debe tener un producto funcional, y así seguir de este modo un tipo de desarrollo iterativo e incremental. Este tipo de metodologías sustituye los antiguos procedimientos que realizaban todas las tareas en serie, para pasar a un tipo de gestión en la que las tareas pueden solaparse. Al no tratarse de un equipo con varios desarrolladores, la metodología utilizada en este proyecto se basa en el Scrum, pero se ve alterada para adaptarse a un equipo de un solo desarrollador y un tutor.

Generalmente los equipos están formados por un número pequeño de participantes para así mejorar en la medida de lo posible la comunicación entre ellos. Dentro del equipo encontramos diferentes roles:

- Product Owner: es el rol encargado de representar al cliente ante el equipo.
- Scrum Master: asegura que el scrum se cumpla debidamente. No es un jefe de equipo, ya que el equipo se autogestiona.
- Equipo de desarrollo: debe disponer de habilidades variadas para completar el desarrollo del producto. Son quienes se encargan de llevar a cabo la propia entrega. Suele ser deseable que el número de participantes no sea excesivo.

Otra característica importante de Scrum son las reuniones. Su función es mantener al equipo informado de todas las novedades acerca del proyecto. Existen multitud de tipos pero los principales son los siguientes:

- Stand-up Meeting: se realiza diariamente y en ellos cada participante explica lo que ha hecho desde el último que se llevó a cabo, qué va a hacer ese día y si se ha encontrado con algún problema.
- Sprint Planning: esta reunión se celebra al inicio de cada sprint con el principal propósito de determinar el objetivo de éste.
- Sprint Review: se realiza al terminar cada sprint y en ella se evalúa el resultado a partir de las tareas que hayan sido finalizadas.





2. Herramientas Software

2.1. GitHub

GitHub [3] es un repositorio web basado en Git y escrito en Ruby que permite tanto el almacenamiento de nuestro código como un control de versiones del mismo.

La versión gratuita de GitHub permite almacenar el código de forma pública, lo cual es suficiente para el propósito de este proyecto.

El software utilizado para sincronizar los archivos locales con el repositorio en la web han sido los clientes propios de GitHub para escritorio en Mac y Windows.

2.1.A. Alternativa1: Bitbucket

Bitbucket [4] ofrece la posibilidad de realizar cinco proyectos privados de forma gratuita. Sin embargo se ha rechazado por el hecho de que GitHub ya había sido utilizado en proyectos anteriores y el proceso de adaptación inicial iba a ser más rápido.

2.1.B. Alternativa2: Sourceforge

Sourceforge [5] es otra alternativa con funcionalidades similares. Del mismo modo, se descartó porque se tenía más conocimiento acerca del uso de GitHub.

2.1.C. Alternativa3: GitLab

GitLab [6] tiene la ventaja de que permite realizar proyectos de forma privada. Del mismo modo que con las alternativas anteriores, se eligió GitHub sobre ella por un mayor conocimiento de la herramienta.

2.2. ZenHub

Zenhub [7] es una extensión para nuestro navegador web que añade funcionalidades extra a la web de GitHub. Con ZenHub podremos incluir un tablero en el que organizar nuestras tareas siguiendo una metodología ágil. Las propias issues de GitHub conformarán las tareas que se añadirán al canvas, mientras que los sprints estarán determinados por los milestones.

También nos permite crear burndown charts a partir de la información añadida al crear cada una de las issues, teniendo en cuenta el tiempo estimado de finalización.

Además de gráficos burndown, incluye la creación de gráficos velocity tracking, que proporcionan una forma más visual de organizar la cantidad de horas que se acumulan en cada sprint.

Gracias al sistema de etiquetas integrado con GitHub, facilita la forma de clasificar las tareas, ya sea en forma de bug, documentación, código, o cualquier tipo de etiqueta que sea desee crear de forma personalizada.

2.2.A. Alternativa1: Trello

Trello [8] era, en este caso, la principal alternativa a ZenHub. Se trata de una plataforma web en la que se dispone de un tablero donde situar las tareas en diferentes columnas. Presentaba los inconvenientes de que para realizar el burndown chart requería de un plugin externo. Además, no se integraba de una forma tan transparente con GitHub como lo hace ZenHub.

2.2.B. Alternativa2: Sprintometer

Sprintometer [9] es una herramienta que nos facilita la división del proyecto en sprints, y es



capaz de realizar los burndown charts, sin embargo no ofrece una interfaz tan intuitiva a modo de canvas como sí lo hacen Trello y ZenHub.

2.3. NetLogo

NetLogo [10] [11] es un lenguaje de programación basado en agentes que dispone de un entorno de desarrollo orientado a la simulación de modelos sociales y naturales.

La potencia de NetLogo reside en los agentes. El programador puede dar órdenes a cada uno de los múltiples agentes, para observar como se comportan independientemente, y en conjunto con el resto.

Dispone de una buena documentación y soporte, además de un sistema de extensiones para poder ampliar las funcionalidades con las que cuenta.

Está orientado principalmente a la docencia, centrando su uso en alumnos y profesores. Sin embargo, tiene potencia suficiente para haber sido utilizado en multitud de publicaciones basadas en investigación.

NetLogo dispone también de una amplia galería de modelos ya desarrollados que proporcionan una base sólida antes de comenzar a desarrollar nuestros propios modelos.

Está basado en Java, lo que permite una disponibilidad multiplataforma.

Durante el desarrollo de este proyecto se ha utilizado la versión 5.3.1 para Mac y Windows.

2.3.A. NetLogo Behaviour Space

NetLogo Behaviour Space [12] es una herramienta incorporada con NetLogo que ofrece la opción de realizar experimentos a partir de cierto modelo. La potencia de esta herramienta reside en que ofrece la posibilidad de ejecutar un modelo un número determinado de veces, almacenando posteriormente los resultados en forma de archivo. Tiene bastante variabilidad en cuanto a la personalización tanto de los parámetros de ejecución, como de los parámetros de salida.

2.3.B. NetLogo Web

Dado que el proyecto se basa en una plataforma web, se requería de un servicio que permitiese el desarrollo del modelo NetLogo en forma de HTML. Para ello NetLogo dispone de una herramienta web [13] que es capaz de exportar una gran variedad de modelos a un formato HTML. No obstante, no está adaptado a numerosas funcionalidades que sí están disponibles en el entorno NetLogo tradicional, entre ellas, la interacción con archivos externos.

2.4. XAMPP

XAMPP [14] es una herramienta orientada a programadores que se están iniciando en el ámbito de los servidores web. Se trata de un servicio que agrupa un conjunto de componentes necesarios para la creación web, instalando todos de forma conjunta para facilitar así su manejo e instalación.

En esencia XAMPP es una distribución de Apache que además del propio servidor añade MariaDB (MySQL), como gestor de bases de datos; y PHP y Perl, como lenguaje de programación en el lado del servidor.

Entre las herramientas que incluye XAMPP destaca PhpMyAdmin. PhpMyAdmin es un gestor web de nuestra base de datos, el cual nos facilitará las tareas de mantenimiento de la misma, así como su configuración.

XAMPP admite también la posibilidad de añadir ciertos complementos, como son, por ejemplo, WordPress, Drupal o Moodle.





Por último XAMPP está disponible en todas las plataformas (Mac, Linux, Windows) de forma gratuita tanto para uso personal como comercial.

2.4.A. Alternativa1: MAMP

MAMP [15] es una alternativa muy similar a XAMPP. Incluye Apache, Nginx, MySQL, PHP, Python y Perl. Tiene la ventaja sobre XAMPP de que incluye Python. Sin embargo no está disponible para Linux, por lo que ha sido rechazado.

2.4.B. Alternativa2: WAMP

WAMP [16] (o Wampserver) es, del mismo modo, una opción muy parecida a las anteriores. Incluye Apache, PHP y MySQL. En este caso también ha sido rechazado por razones de disponibilidad por sistema operativo, ya que en este caso es exclusivo de Windows.

2.5. HTML

HTML [17] (HyperText Markup Language) es el estándar de referencia en la realización de páginas Web. Se trata de un lenguaje basado en etiquetas que establecen la disposición de los contenidos de una página web.

Se ha utilizado HTML5 por ser la versión más actual, ya que los navegadores están orientados a adaptarse a ella.

2.6. JavaScript

Como lenguaje de programación en el lado del cliente se ha utilizado JavaScript [18]. JavaScript es un lenguaje orientado a objetos que se utiliza generalmente en programación web del lado del cliente, aunque también se puede usar en aplicaciones de escritorio (sobretudo en widgets).

Gracias a estar en el lado del cliente, permite multitud de funcionalidades visuales en el navegador.

2.6.A. jQuery

Jquery [19] es una biblioteca JavaScript orientada al manejo de eventos e interacciones en nuestro código HTML. Facilita ciertas tareas, suponiendo un ahorro de código y esfuerzo. Proporciona también la posibilidad del uso de Ajax, que se tratará en profundidad a continuación.

Se trata de software libre, con libertad de uso en proyectos tanto públicos como privados.

Se ha utilizado la última versión hasta el momento, la versión 3.1.1.

2.6.B. Ajax

Ajax [20] (Asynchronous JavaScript and XML) es una técnica de programación web basada en una interacción con el servidor de forma asíncrona y más transparente al usuario, intentando evitar la recarga de la página cuando se realiza una petición.

Cabe destacar que Ajax no es un lenguaje de programación, ni una nueva herramienta web, es simplemente una técnica que aprovecha las capacidades de los navegadores para conseguir una serie de interacciones mucho más limpias de cara al usuario.

2.7. PHP

PHP [21] (PHP: Hypertext Preprocesor) es un lenguaje de programación de código abierto



destinado al desarrollo web. Su potencia reside en que puede ser incrustado dentro de un código HTML utilizando las etiquetas “<?php” de inicio y “?” de finalización. No obstante, también puede ser utilizado en scripts independientes del html siempre que hagamos referencia a ellos.

PHP es un lenguaje que se ejecuta del lado del servidor, esto quiere decir que, a diferencia de JavaScript, no se ejecutará en el navegador del usuario, sino que se deberá hacer una petición al servidor de la página web para ser llevado a cabo en ese lado.

Ofrece una gran disponibilidad en cuanto a sistemas operativos y navegadores, siendo soportado por la gran mayoría, lo que ha ocasionado que sea uno de los lenguajes más populares en la actualidad [22].

2.8. CSS

CSS [23] (Cascading Style Sheets) es un lenguaje de programación destinado a la modificación estética de los elementos de una página web HTML o XML. CSS actúa sobre la renderización del elemento para definir cómo debe ser presentado al usuario.

La base del lenguaje CSS reside en los estilos que definen la forma de presentar un documento HTML o XML. CSS especifica esos estilos en scripts independientes que pueden ser utilizados en varios documentos web simultáneamente. De esta forma, si queremos realizar un cambio de estilo global, en lugar de alterar cada uno de los documentos, variaremos un único fichero CSS.

2.9. Bootstrap

Un principio a tener en cuenta en programación es que hay muchas funcionalidades que ya están creadas y no merece la pena perder el tiempo en rehacerlas de nuevo, es decir, no deberíamos tratar de reinventar la rueda. Bootstrap [24], [25] es un framework que nos ofrece una gran cantidad de funcionalidades y estilos en forma de librerías que nos ahorran una gran cantidad de tiempo de desarrollo.

Su gran fuerte es que ofrece una forma muy simple de crear páginas “responsive”, es decir, que se adapten al dispositivo en el que se están mostrando en cada momento, según su tamaño. Esta es una gran ventaja en una era en la que disponemos de una gran cantidad de tamaños y resoluciones de pantalla debido al uso de smartphones, tablets o smartwatches añadidos a los ya establecidos ordenadores portátiles o monitores de sobremesa.

Además de la propia adaptación a diferentes dispositivos, también proporciona una gran variedad de plantillas, estilos, y elementos de la web ya desarrollados a los que podremos acudir como modelo, para no tener la necesidad de desarrollarlos desde cero, y de esta manera poder ahorrar mucho tiempo de trabajo y aprendizaje.

Se ha utilizado la versión 3.3.7 por ser la última versión estable en el momento de desarrollo del proyecto.

2.10. SublimeText

SublimeText [26] es un editor de texto, inicialmente derivado de Vim, que finalmente desarrolló un variante propia y ahora mismo cuenta con su versión número 3. Está disponible para una gran cantidad de lenguajes de programación, lo que ha supuesto una ventaja en este proyecto, ya que se ha podido utilizar con prácticamente todos los lenguajes requeridos.

A simple vista parece un editor de texto que trabaja sobre un solo archivo, sin embargo es capaz de actuar sobre un directorio completo, adquiriendo así ciertos tintes de un entorno de desarrollo más avanzado que un simple editor de texto.



En este caso se ha utilizado la versión 3. SublimeText es un software de pago, sin embargo



dispone de una versión de evaluación con funcionalidad plena sin fecha de finalización.

2.10.A. Plugin Emmet

Se ha añadido el plugin Emmet [27] a SublimeText. Emmet es una herramienta que facilita la escritura tanto de código HTML como CSS. Su objetivo principal es conseguir una sintaxis correcta además de una mayor comodidad y un ahorro considerable de tiempo. Añade una serie de comandos que consiguen crear ciertas líneas de código a base de abreviaturas e incluso bloques de código con un simple comando.

2.10.B. Alternativa1: Brackets

Brackets [28] es un editor enfocado a desarrollo web (HTML, CSS y JavaScript) apoyado por Adobe y de código libre. Tiene un gran soporte que actualiza el editor cada muy poco tiempo. Tiene la ventaja de ser gratuito y de poseer una herramienta que permite ver los cambios en la web sin recargar el navegador. Por otro lado, el soporte para PHP depende de extensiones independientes, por esta razón se ha elegido SublimeText sobre Brackets.

2.10.C. Alternativa2: Atom

Atom [29] es un editor de texto open source disponible en Linux, Mac y Windows que soporta gran cantidad de lenguajes, entre los que se encuentran PHP, HTML, CSS, JavaScript o SQL.

Existe la posibilidad de acceder al proyecto de Atom directamente sobre GitHub. Está orientado a que el propio usuario modifique el código del editor si lo desea. Su principal ventaja reside en esa personalización, pero ya que no iba a ser utilizada en profundidad en este caso, se decidió contar con SublimeText en su lugar.

2.10.D. Alternativa3: Eclipse

Se trata de una herramienta [30] diferente a las anteriores. En estos casos se trataba de editores de texto que intentan abarcar el máximo número de lenguajes posible, especializándose en programación web. El caso de Eclipse es diferente, se trata de un entorno completo de desarrollo que se centra algo más en un lenguaje específico en cada caso.

Una de las ventajas más importantes es que permite la sincronización con GitHub en el propio entorno de desarrollo de una forma muy sencilla. Se intentó estudiar esta alternativa por haber sido utilizada en bastantes casos a lo largo de toda la carrera. Finalmente, se concluyó que las otras herramientas se adaptaban más a las necesidades de este proyecto.

2.11. Mendeley

Mendeley [31] es un gestor de referencias bibliográficas que simplifica el almacenamiento y la forma de compartir documentos relacionados con la investigación. Consta de varios componentes: una página web, una extensión de navegador, una aplicación de escritorio y un plugin para nuestro editor de texto.

Además de la propia gestión de referencias, tiene ciertos aspectos que emulan a una red social para investigadores, con la intención de crear una comunidad a raíz de las referencias entre ellos en sus artículos.

2.11.A. Alternativa: Zotero

Zotero [32] es una herramienta muy similar a Mendeley. Es libre y abierto. Del mismo modo que Mendeley dispone de sitio web, extensión de navegador, aplicación de escritorio y plugin para editores de texto.



Al tratarse de dos herramientas muy similares no existe una diferencia realmente significativa, así que por recomendación de usuarios se ha decidido elegir Mendeley.

2.12. LibreOffice Writer

Para llevar a cabo la propia memoria del proyecto se ha seleccionado la herramienta Writer [33] de LibreOffice. Se ha elegido por encima de OpenOffice Writer y de Word por disponer a la vez de una plantilla del proyecto en su formato y un plugin de Mendeley que funciona correctamente y es fácil de instalar. Además se trata de una herramienta gratuita.

2.12.A. Alternativa1: OpenOffice Writer

Writer [34] es el procesador de texto del paquete de herramientas de ofimática propiedad de Apache, denominado OpenOffice. Tiene la ventaja de que su descarga es gratuita. Además existe una plantilla del proyecto adaptada en el formato propio de Writer (odt). Sin embargo ha sido descartada por no disponer de un plugin de Mendeley de forma fácil y accesible.

2.12.B. Alternativa2: Microsoft Word

Se trata del procesador de textos del paquete de herramientas ofimáticas de Microsoft, Microsoft Office [35]. En este caso el software es de pago, pero se dispone de una licencia de Office 365 por ser alumno de la Universidad de Burgos. El plugin de Mendeley es mucho más fácil de instalar y presenta menos problemas que en OpenOffice. Sin embargo ha sido rechazado por no disponer de una plantilla del proyecto propia para su formato.

2.12.C. Alternativa3: LaTeX

LaTeX [36] es una alternativa de editor de texto bastante diferente a las opciones anteriores. Está basado en la inclusión de etiquetas en el texto que definen sus características. Este tipo de orientación ofrece una mayor libertad en el desarrollo de un documento si se aprende a usar de forma correcta.

Tiene la ventaja de que dispone de una plantilla para el proyecto en formato propio. Por un lado ofrece un potencia mayor, pero por otro lado presenta una curva de aprendizaje que los otros editores no plantean. Por esa razón LibreOffice Writer ha sido finalmente el seleccionado.





V - ASPECTOS RELEVANTES DEL DESARROLLO DEL PROYECTO

La experiencia del desarrollo del proyecto se describe a través de tres bloques diferenciados. El primero de ellos es el modelo de NetLogo destinado, en última instancia, a que los alumnos prueben sus estrategias. El segundo de estos bloques es el modelo global de NetLogo, aquel orientado a realizar el torneo final entre todos los alumnos. El tercero y último está centrado en los elementos relacionados con la propia web, es decir, el servidor y la aplicación web en sí misma. Podría diferenciarse un bloque paralelo a los demás centrado en la investigación, en el cual no profundizaremos ya que puede verse en detalle en los apartados de conceptos teóricos y trabajos relacionados.

Antes de pasar a analizar los tres apartados generales, veamos cómo se ha enfocado el proyecto desde el momento inicial. Se buscaba una plataforma en la que poder realizar torneos del estilo del realizado por Axelrod [1]. Para ello era necesario crear los modelos con una herramienta que facilitara su implementación, se seleccionó NetLogo por cumplir las expectativas. Como referencia inicial se usó el modelo desarrollado por Janssen [2]. Inicialmente se pensaba proponer a los alumnos utilizar la herramienta de NetLogo web para poner a prueba sus estrategias independientemente de nuestra plataforma. El cambio principal sobre el planteamiento inicial ha sido incluir el propio modelo en la web, gracias a las facilidades que proporciona NetLogo web para exportar los modelos en HTML. Otro aspecto que se buscaba en el planteamiento inicial era que los profesores pudiesen fijar los parámetros de payoff antes de realizar el torneo de forma personalizada, de esta forma podían plantear varias posibilidades dentro de los torneos de tipo 2 x 2. Además, se deseaba que se pudiesen celebrar varios torneos a la vez dentro de la plataforma.

La forma de orientar el proyecto ha sido buscar siempre la funcionalidad, permitiendo con ello probar la aplicación desde los primeros momentos, solamente con NetLogo, hasta los últimos, desde la web. Se ha preferido dejar para el final ciertas funcionalidades no esenciales de la web para poder profundizar primero en comportamiento del torneo.

1. *Modelo individual*

Pasemos ahora a analizar el primero de los tres puntos principales en los que se puede dividir en enfoque del proyecto. Como ya se ha especificado, este punto está referido al modelo del torneo con el que los participantes pueden probar sus estrategias. El desarrollo del modelo inicialmente no generó problemas más allá de un cambio general en la forma de gestionar los participantes. Se diseñó e implementó el funcionamiento del torneo y varias estrategias contra las que se puede probar la estrategia del alumno, además de una gráfica con la que mostrar los resultados, sin problemas reseñables. Respecto a la forma de gestionar los participantes, al principio se había llevado a cabo sin tener en cuenta las facilidades que ofrece NetLogo con sus agentes (turtles), lo que daba como resultado un código menos eficiente. Para mejorar este aspecto se volvieron a estructurar los participantes como agentes y se utilizó la primitiva de NetLogo runresult para una optimización de las estrategias. El primer problema a sortear se presentó cuando, al intentar utilizar este modelo en NetLogo web, no se podía efectuar la ejecución. El problema venía a raíz de que NetLogo web tiene ciertas funcionalidades limitadas respecto a su versión de escritorio. Por ello, se tuvo que descartar el uso de runresult utilizando un código menos simplificado, pero con la misma funcionalidad.

```
ask players [
  set decision runresult my-strategy
  set past-decisions lput decision past-decisions ]
ifelse my-strategy = "alwaysCooperate"[set decision alwaysCooperate set past-decisions lput decision past-decisions]
[ifelse my-strategy = "alwaysDefect"[set decision alwaysDefect set past-decisions lput decision past-decisions]
  [ifelse my-strategy = "titForTat"[set decision titForTat set past-decisions lput decision past-decisions]
    [ifelse my-strategy = "titForTwoTats"[set decision titForTwoTats set past-decisions lput decision past-decisions]
      [ifelse my-strategy = "friedman"[set decision friedman set past-decisions lput decision past-decisions]
        [ifelse my-strategy = "joss"[set decision joss set past-decisions lput decision past-decisions]
          [ifelse my-strategy = "estrategiaAlumno"[set decision estrategiaAlumno set past-decisions lput decision past-decisions]
            [if my-strategy = "randomSt"[set decision randomSt set past-decisions lput decision past-decisions]]
          ]
        ]
      ]
    ]
  ]
]
```

Illustration 2: Solución ante la falta de soporte de runresult por parte de NetLogo web.

Sí surgió un problema que llevó bastante trabajo solucionar en el momento de incluir el modelo de NetLogo web, exportado en HTML, a la aplicación web. El problema era que ese código exportado no cumplía los estándares de HTML5 como el resto de la web. Esto provocaba que la apariencia del modelo no fuese la correcta e impidiese la normal utilización de la herramienta. En principio la solución era sencilla: modificar el código para que cumpla los estándares. Sin embargo ese código estaba generado por NetLogo web. NetLogo web incluye en sus modelos exportados toda su tecnología para evitar al usuario su instalación. Esto provocaba un código de difícil comprensión con una gran cantidad de líneas. Finalmente, modificando algunos estilos y una plantilla que generaba parte del modelo, se consiguió rectificar ese código para que el aspecto en la web permaneciese como debería .

158	159	-	<div id="netlogo-model-container" style="background-color: #F8F8FF;padding-top: 20px;padding-right: 20px;padding-left: 20px;padding-bottom: 20px;">	
	159	+	<div id="netlogo-model-container" style="background-color: #F8F8FF;padding-top: 20px;padding-right: 20px;padding-left: 20px;padding-bottom: 20px;">	
159	160		<script>/*!	
160	161		* jQuery JavaScript Library v3.1.1	
161	162		* https://jquery.com/	
		✖	@@ -59358,10 +59359,10 @@ <h3 id="alert-title">Error</h3>	
59358	59359		View.prototype.transformCanvasToWorld = function(world, canvas, ctx) {	
59359	59360		var ref;	
59360	59361		this.quality = Math.max((ref = window.devicePixelRatio) != null ? ref : 2, 2);	
59361		-	this.maxpxcor = world.maxpxcor != null ? world.maxpxcor : 25;	
59362		-	this.minpxcor = world.minpxcor != null ? world.minpxcor : -25;	
59363		-	this.maxpycor = world.maxpycor != null ? world.maxpycor : 25;	
59364		-	this.minpycor = world.minpycor != null ? world.minpycor : -25;	
	59362	+	this.maxpxcor = (world.maxpxcor != null ? world.maxpxcor : 25)/2;	
	59363	+	this.minpxcor = (world.minpxcor != null ? world.minpxcor : -25)/2;	
	59364	+	this.maxpycor = (world.maxpycor != null ? world.maxpycor : 25)/2;	
	59365	+	this.minpycor = (world.minpycor != null ? world.minpycor : -25)/2;	
59365	59366		this.patchsize = world.patchsize != null ? world.patchsize : 9;	
59366	59367		this.wrapX = world.wrappingallowedin;	
59367	59368		this.wrapY = world.wrappingallowedin;	
		✖	@@ -60629,7 +60630,7 @@ <h3 id="alert-title">Error</h3>	
60629	60630		}());	
60630	60631		};	
60631	60632			
60632		-	template = "<div class=\"netlogo-model\" style=\"min-width: {{width}}px;\">	tabindex=\"1\" on-keydown=\"checkA
	60633	+	template = "<div class=\"netlogo-model\" style=\"min-width: {{width}}px;\">	tabindex=\"1\" on-keydown=\"checkA
60633	60634			

Illustration 3: Solución al problema de compatibilidad del modelo NetLogo web con html5

2. Modelo global



El segundo de los bloques ya mencionados es el relacionado con el modelo del torneo final. El



principal tema a tratar en este aspecto es la gestión de las estrategias de los alumnos. NetLogo no soporta de forma nativa interacciones con bases de datos. A raíz de este hecho se planteaba la decisión de cómo almacenar los participantes, sus estrategias, los resultados, etc. La alternativa tomada finalmente ha consistido en almacenar estos elementos en ficheros en el servidor. El proceso consiste en que, cuando el alumno entrega su estrategia a través de la web, se guarda su modelo en un archivo (.nlogo) en el servidor. A continuación, cuando se lanza el torneo desde el modelo global de NetLogo, éste extrae la estrategia del modelo anterior en un fichero (.nls) que será importado a posteriori por el modelo del torneo global. Esta solución también facilita la forma de proporcionar al alumno una posibilidad de descargar su estrategia incluyendo el modelo de prueba completo. Con los resultados y los participantes sucede algo similar. Por cada participante, al comienzo de la ejecución del torneo, se escribe una línea en el fichero de participantes para identificarlo, así el modelo NetLogo puede recibir la información. Los resultados llevan a cabo el mismo proceso en sentido inverso, es decir, NetLogo exporta los resultados de la ejecución en un fichero, que es posteriormente leído por el servidor y se almacena su información en la base de datos.

Otro punto a tener en cuenta relacionado con el modelo global es la forma de ejecutarlo. No es posible ejecutar NetLogo cargando el modelo necesario desde el servidor, ya que eso incluiría funciones gráficas que no son necesarias para nuestro propósito. NetLogo ofrece un entorno en el que realizar experimentos con los modelos llamado “behaviour space”, además de un modo de ejecución en fondo por línea de comandos que no requiere de interfaz gráfica (headless). Así, se ha utilizado un fichero ejecutable que utiliza Java para lanzar el modelo que deseemos, con los parámetros que se quieran fijar para tal experimento, a través de ciertos archivos .jar que ofrece NetLogo. Esto permite la realización del torneo desde un servidor en el que no es necesaria la instalación de NetLogo, con la existencia de los archivos .jar y tener Java instalado es suficiente.

3. Aplicación web

Una vez estudiado el enfoque adoptado para los modelos de NetLogo, pasemos al tercer punto principal: la plataforma web sobre la que se sustenta el proyecto. Consta de varios apartados, se ha desarrollado una página principal, en la que se muestra una breve descripción de su contenido, así como los torneos activos y su número de participantes. Desde este punto se ha buscado permitir utilizar la barra superior para logearnos en la página o, si no tenemos una cuenta, ir a la página de registro para crearla. Una vez logeados, si somos profesores podremos finalizar los torneos (solo aquellos que hayamos creado nosotros), crear un torneo nuevo, unirnos a uno, consultar los resultados de otros ya celebrados o ir a nuestro perfil a comprobar nuestros resultados anteriores. Veamos más en profundidad algunas de estas funcionalidades.

El proceso de vida desarrollado para un torneo ha sido el siguiente. En primer lugar, un usuario que previamente se haya registrado como profesor debe crear el torneo. Esto añadirá un nuevo torneo en base de datos con los parámetros fijados por el profesor y asignará el id de ese torneo a la sala en la que se haya creado. Posteriormente, los participantes se unirán al torneo y observarán el modelo de NetLogo. En este caso, surgía la decisión acerca de cómo plantear la participación de los alumnos permitiéndoles poner a prueba sus estrategias. En enfoque adoptado finalmente fue ofrecer el código del modelo a los alumnos con un método que pueden modificar. Ese método será su estrategia ante el torneo, y el que se guardará posteriormente. Este enfoque permite al usuario probar su estrategia para así comprobar tanto su rendimiento como su correcto funcionamiento.

El otro aspecto que merece consideración en cuanto a la forma de enfocararlo es la finalización del torneo. Como ya se ha explicado anteriormente, NetLogo necesita recibir la información por



ficheros. Teniéndolo en cuenta, la información de un torneo desde la web se guarda siempre en base de datos. Lo que hace la finalización del torneo es escribir esa información necesaria en ficheros para que NetLogo pueda leerlos y, una vez que el modelo NetLogo ha terminado su ejecución, reciba la información que ha sido exportada a ficheros y la almacena en la base de datos.

Finalicemos con el análisis del enfoque acerca de la forma de gestionar los tipos de usuarios. Era necesario ofrecer dos tipos de registro: como alumno (participante) y como profesor (creador de los torneos). Con el fin de limitar el tipo de funcionalidad a la que puede acceder cada uno, se deseaba ocultar cierta información a aquellos usuarios que no tuviesen los permisos suficientes para realizar ciertas acciones. Para ello se ha utilizado jQuery junto con ajax para ocultar la posibilidad de crear y finalizar torneos a los alumnos, ellos simplemente no pueden ver esas opciones, mientras que aquellos usuarios registrados como profesores pueden acceder a esas funcionalidades sin ningún inconveniente.

3.1. Servidor web

Como breve anotación final al apartado de la aplicación web vamos a analizar el enfoque llevado a cabo en el servidor. Se ha buscado evitar que el servidor requiera de la tecnología de NetLogo a través del uso de las funcionalidades web y de las bibliotecas que nos proporciona el propio NetLogo. Con esto se buscaba poder utilizar la aplicación a través de un servidor sencillo de dominio gratuito, para así, en caso de querer poner en marcha la aplicación en un futuro, tener la mayor facilidad posible.

Finalmente, en busca de la sencillez y de una forma de probar el proyecto más simple en el entorno de un trabajo de fin de grado, se ha utilizado un servidor local para simular el servidor final que tendría la aplicación. Todo esto sin olvidar el objetivo de mantener el servidor sin necesidad de una instalación de NetLogo en el mismo. Lo que sí se ha requerido finalmente es la instalación de Java para permitir la ejecución de los .jar necesarios para el funcionamiento del modelo de NetLogo en el servidor.





VI - TRABAJOS RELACIONADOS

Existe una gran variedad de artículos científicos orientados al estudio de la enseñanza a través de herramientas de simulación y juegos. Primero veremos algunos de los artículos que se centran en este tipo de estudios. A continuación nos centraremos en aquellos que profundicen en temas similares al propósito de este proyecto, es decir, la utilización de herramientas de simulación destinadas a la enseñanza de temas relacionados con la teoría de juegos o con conceptos más cercanos a la rama de la economía.

En primer lugar, es necesario hacer mención a Huang y Despande en su trabajo “*Simulation Games in Engineering Education: A State-of-the-Art Review*” [1]. En este artículo se plantea que las tecnologías de la información han supuesto una influencia destinada al cambio del modelo educacional tradicional. Debido a ello, se analizan una serie de herramientas utilizadas en varios campos diferentes dentro de la ingeniería. A continuación veremos algunos de los estudios que se tienen en cuenta en el artículo.

El primero de ellos, “The Effectiveness of Games for Educational Purposes: A Review of Recent Research” [2] lleva a cabo un análisis de la efectividad del uso de este tipo de herramientas para la enseñanza. En él se realiza una comparación de los métodos tradicionales con aquellos en los que se utilizan juegos o simulaciones. Tras analizar numerosos estudios sobre el tema, concluye que en el 72% de los casos en los que se aplica esta nueva técnica, los resultados son al menos tan buenos como con las metodologías tradicionales. Mientras que en el 22% de los casos, presentan unos resultados mejores.

Una vez visto un análisis general del tema, podemos pasar a tratar brevemente algunos de los estudios específicos realizados en varios campos de la enseñanza.

- “*Using a Simulation Game Approach to Teach Enterprise Resource Planning Concepts*” [3] propone un tipo de enseñanza de gestión de recursos empresariales haciendo uso de una herramienta de simulación. Se plantea a los alumnos la gestión de su propia compañía a través de la toma de decisiones, analizando la información que plantea el simulador. El artículo concluye que los alumnos que han participado en la simulación parecen haber asimilado mejor los conceptos teóricos.
- “*The development of an interactive game-based tool for learning surgical management*” [4] propone la ampliación de un artículo anterior en el que se demostraba la eficacia de un juego de mesa en la enseñanza en el campo de la medicina [5]. En este artículo se propone dar un paso más, utilizando en su lugar un simulador de ordenador. El artículo concluye que la herramienta ha demostrado su eficacia en los casos planteados por el simulador, y que puede ser ampliado a otros casos médicos.
- “*Using Computer-Based Simulation Exercises to Teach business Ethics*” [6] realiza un estudio a través del planteamiento de dilemas éticos a estudiantes, utilizando para ello un simulador. Así pretende dotar a los docentes de un medio y un enfoque más óptimos en el proceso de enseñanza de la materia.
- “*Improving Visualization Skills of Engineering Graphics Students Using Simple JavaScript Web Based Games*” [7] es un artículo en el que se proporciona a una serie de estudiantes varias herramientas en forma de juegos y simuladores para, así, poner a prueba su efectividad en el proceso de aprendizaje de varios aspectos, todos ellos relacionados con la ingeniería gráfica. Como en los casos anteriores, se concluye una mejora de rendimiento en los exámenes así como unas buenas valoraciones de los estudiantes en las



encuestas.

- En “*Designing Computer Games to Help Physics Students Understand Newton's Laws of Motion*” [8] se utiliza un entorno de simulación para familiarizar a los alumnos con los conceptos de las leyes de Newton y su implicación. Las conclusiones, a raíz de un estudio controlado, exponen que este simulador ha provocado que los alumnos mejoren sus capacidades para resolver problemas dinámicos.

Dentro del campo de la informática también encontramos artículos sobre el tema. Trataremos, a continuación, dos ejemplos:

- “*Simulation game for teaching communications protocols*” [9]. En este artículo se plantea dotar a los profesores de un simulador con el que explicar los aspectos algorítmicos de los diferentes protocolos de comunicación. Los investigadores concluyen que la utilización del simulador ha supuesto una mejora tanto en el rendimiento de los alumnos, como del interés de los mismos en la materia.
- “*Simse: an interactive simulation game for software engineering education*” [10]. Este artículo argumenta como base de su estudio que los alumnos de ingeniería del software tiene una buena base teórica pero no pueden poner en práctica estos conceptos. Para solucionar esta carencia, desarrollan Simse, una herramienta de simulación con la que plantear situaciones típicas de un proceso de ingeniería del software a los alumnos, con el fin de poner a prueba sus conocimientos teóricos de una forma práctica. En un artículo posterior [11] se realiza un análisis de la herramienta en estudios controlados, los cuales concluyen que los resultados son satisfactorios para el rendimiento de los alumnos.

Finalmente, acercándonos al campo de estudio más relacionado con el proyecto en cuestión, encontramos varios artículos que profundizan en el tema.

- El primero de los ejemplos no está relacionado con el uso de simuladores, como los anteriores, sino que sirve como ejemplo de la efectividad de los experimentos en el aula para la ilustración ciertos conceptos. En el caso de “*The impact of classroom experiments on the learning of economics: an empirical investigation*” [12] se realiza un breve experimento que, por su facilidad, puede llevarse a cabo en cualquier clase. El artículo concluye que el grupo que había realizado ese experimento obtuvo mejores resultados en tests de múltiple respuesta que aquellos que formaban parte del grupo control.
- El segundo artículo se ha seleccionado por tratar un tema común: el desarrollo de una herramienta destinada a la docencia en un campo de estudio bastante relacionado. También se utiliza NetLogo como tecnología base para el modelo. Se trata de “*Quality Uncertainty and Market Failure: An Interactive Model to Conduct Classroom*” [13] y en él se lleva a cabo el desarrollo de una versión interactiva de la herramienta creada en un artículo anterior [14]. En este artículo previo se había estudiado, a través de un modelo basado en agentes, el efecto de la incertidumbre en la calidad, dentro de un entorno de mercado sin intercambio asimétrico de información.

Los artículos analizados anteriormente evidencian que la herramienta desarrollada en este proyecto puede ser muy útil para complementar los conceptos explicados en clase acerca de la teoría de juegos.





VII - CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

1. Conclusiones

Podemos finalizar esta memoria concluyendo que, en general, se han alcanzado los objetivos planteados al inicio del proyecto satisfactoriamente. Además se han ampliado conocimientos en varios campos y se han aplicado técnicas y herramientas aprendidas a lo largo del grado.

Se ha logrado desarrollar una plataforma docente que cumple las especificaciones que se fijaron en su planteamiento inicial.

Ha resultado beneficioso profundizar en conocimientos como la teoría de juegos, el desarrollo web, la gestión de proyectos, etc.

Los conocimientos adquiridos a lo largo de la carrera se han visto reflejados a lo largo del proyecto, abarcando varias disciplinas diferentes.

El trabajo con artículos científicos y algunos libros ha ayudado a familiarizarse con este tipo de información. También se han mejorado los conocimientos acerca de cómo citar y referenciar correctamente estos contenidos.

Finalmente, se ha abierto una puerta a mejorar esta plataforma en proyectos futuros con esta primera versión. A continuación veremos los posibles caminos a tomar para aumentar la funcionalidad de este proyecto.

2. Líneas de trabajo futuras

La evolución lógica de este proyecto, teniendo en cuenta la evolución del torneo de Axelrod [1], sería continuar con una versión evolutiva del mismo. El objetivo sería crear poblaciones de estrategias que interaccionarían unas con otras, y estudiar cómo evolucionan en el tiempo y bajo qué condiciones algunas de esas estrategias son evolutivamente estables. Esta mejora aumentaría la complejidad del proyecto, proporcionando una perspectiva diferente a los torneos, permitiendo también evaluar su funcionamiento comparándolo con el del torneo de Axelrod.

Otra línea de trabajo muy interesante sería la integración de la herramienta dentro de la plataforma de Moodle. Dado que se trata de una aplicación docente, sería mucho más accesible para los alumnos dentro de este sistema. Además, daría más opciones a los profesores aprovechando las funcionalidades que ya tiene Moodle. También simplificaría y mejoraría el sistema de usuarios.

Con el objetivo de un acercamiento inicial de los torneos a los alumnos, sería interesante plantear un tipo de torneo en la plataforma que enfrente a dos alumnos directamente, sin desarrollar una estrategia previa, tomando ellos las decisiones en cada turno. Así, podría realizarse, por ejemplo, el dilema del prisionero iterado y estudiar las elecciones de los participantes. Esta opción permitiría realizar experimentos un tanto diferentes y facilitaría la comprensión de las mecánicas antes de presentar a los alumnos el torneo convencional.

Además de las funcionalidades más grandes comentadas anteriormente, se podría mejorar la herramienta con pequeños cambios en la funcionalidad que supongan una mejora en su utilización:

- Mejorar el diseño de la web.
- Añadir clave a los torneos creados.
- Proporcionar un método de búsqueda con filtros para los torneos ya finalizados.
- Sistema de recuperación de contraseñas.



- Algunos avisos de error podrían mostrarse en la misma página, sin enviarnos a otra perdiendo la interfaz.
- Internacionalizar la web con otros idiomas, especialmente el inglés.

VIII - REFERENCIAS

Bibliografía

Capítulo III: Conceptos teóricos.

- [1] R. M. Axelrod, *The evolution of cooperation*. Basic Books, 2006.
- [2] J. Pérez, J. L. J. P. Jimeno, and E. Cerdá Tena, *Teoría de juegos*. Pearson Educación, 2003.
- [3] P. Walker, “Chronology of Game Theory.” [Online]. Available: http://www.econ.canterbury.ac.nz/personal_pages/paul_walker/gt/hist.htm. [Accessed: 17-Dec-2016].
- [4] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton University Press, 2007.
- [5] J. F. Nash, “Equilibrium Points in N-Person Games,,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 36, no. 1, pp. 48–9, Jan. 1950.
- [6] J. F. Nash, “The Bargaining Problem,” *Econometrica*, vol. 18, no. 2, p. 155, Apr. 1950.
- [7] J. Nash, “Non-Cooperative Games,” *Ann. Math.*, vol. 54, no. 2, p. 286, Sep. 1951.
- [8] J. F. Nash, Nash, and John, “Two-Person Cooperative Games,” *Econometrica*, vol. 21, no. 1, pp. 128–140, 1953.
- [9] “- Google Académico.” [Online]. Available: <https://scholar.google.es/scholar?hl=es&q=game+theory&btnG=&lr=&oq=game+th>. [Accessed: 17-Dec-2016].
- [10] J. Maynard Smith, *Evolution and the theory of games*. Cambridge University Press, 1982.
- [11] “Rock, Paper, Scissors.” [Online]. Available: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/1998-99/game-theory/psr.html>. [Accessed: 18-Dec-2016].
- [12] “Symmetric Game - Game Theory .net.” [Online]. Available: <http://www.gametheory.net/dictionary/Games/SymmetricGame.html>. [Accessed: 17-Dec-2016].
- [13] “Nash Equilibrium.” [Online]. Available: <http://www.investopedia.com/terms/n/nash-equilibrium.asp>. [Accessed: 18-Dec-2016].
- [14] R. Axelrod, “Effective Choice in the Prisoner’s Dilemma,” *Source J. Confl. Resolut.*, vol. 24, no. 1, pp. 3–25, 1980.
- [15] R. Axelrod, “More Effective Choice in the Prisoner’s Dilemma,” *Source J. Confl. Resolut.*,





vol. 24, no. 3, pp. 379–403, 1980.

- [16] R. M. Axelrod, *The complexity of cooperation : agent-based models of competition and collaboration*. Princeton University Press, 1997.
- [17] “Axelrod’s Tournament.” [Online]. Available: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/1998-99/game-theory/axelrod.html>. [Accessed: 18-Dec-2016].
- [18] “Teoría de Juegos - Wikipedia,” 2011. [Online]. Available: https://es.wikipedia.org/w/index.php?title=Teor%C3%ADa_de_juegos&oldid=95296572 [Accessed: 17-Dec-2016].

Capítulo IV: Técnicas y herramientas

- [1] K. Beck *et al.*, “Manifesto for Agile Software Development,” 2001. [Online]. Available: <http://agilemanifesto.org/>. [Accessed: 14-Dec-2016].
- [2] D. Y Soportado Por, K. Schwaber, and J. Sutherland, “La Guía de Scrum La Guía Definitiva de Scrum: Las Reglas del Juego,” 2013.
- [3] “GitHub - How people build software.” [Online]. Available: <https://github.com/>. [Accessed: 15-Dec-2016].
- [4] “Bitbucket | The Git solution for professional teams.” [Online]. Available: <https://bitbucket.org/>. [Accessed: 15-Dec-2016].
- [5] “SourceForge - About.” [Online]. Available: <https://sourceforge.net/about>. [Accessed: 15-Dec-2016].
- [6] “Code, test, and deploy together with GitLab open source git repo management software | GitLab.” [Online]. Available: <https://about.gitlab.com/>. [Accessed: 15-Dec-2016].
- [7] “ZenHub - Agile GitHub Project Management Software.” [Online]. Available: <https://www.zenhub.com/>. [Accessed: 15-Dec-2016].
- [8] “Trello.” [Online]. Available: <https://trello.com/>. [Accessed: 15-Dec-2016].
- [9] “Sprintometer - Scrum & XP project tracking | Website of Scrum tool Sprintometer.” [Online]. Available: <http://sprintometer.com/>. [Accessed: 15-Dec-2016].
- [10] “NetLogo Home Page.” [Online]. Available: <https://ccl.northwestern.edu/netlogo/>. [Accessed: 15-Dec-2016].
- [11] “Manual de Netlogo.” [Online]. Available: <https://sites.google.com/site/manualnetlogo/vistas-en-netlogo-1>.
- [12] “NetLogo 5.3.1 User Manual: BehaviorSpace Guide.” [Online]. Available: <https://ccl.northwestern.edu/netlogo/docs/behaviorspace.html#advanced>. [Accessed: 15-



Dec-2016].

- [13] “NetLogo Web.” [Online]. Available: <http://www.netlogoweb.org/>. [Accessed: 15-Dec-2016].
- [14] “XAMPP Installers and Downloads for Apache Friends.” [Online]. Available: <https://www.apachefriends.org/es/index.html>. [Accessed: 15-Dec-2016].
- [15] “MAMP & MAMP PRO.” [Online]. Available: <https://www.mamp.info/en/>. [Accessed: 15-Dec-2016].
- [16] “WampServer, la plate-forme de développement Web sous Windows - Apache, MySQL, PHP.” [Online]. Available: <http://www.wampserver.com/en/>. [Accessed: 15-Dec-2016].
- [17] “Introduction to HTML.” [Online]. Available: http://www.w3schools.com/html/html_intro.asp. [Accessed: 15-Dec-2016].
- [18] “About JavaScript - JavaScript | MDN.” [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript?redirectlocale=en-US&redirectslug=JavaScript%2FAbout_JavaScript. [Accessed: 15-Dec-2016].
- [19] “jQuery.” [Online]. Available: <https://jquery.com/>. [Accessed: 15-Dec-2016].
- [20] “AJAX Introduction.” [Online]. Available: http://www.w3schools.com/xml/ajax_intro.asp. [Accessed: 15-Dec-2016].
- [21] “PHP: Documentation.” [Online]. Available: <http://php.net/docs.php>. [Accessed: 15-Dec-2016].
- [22] “The 2016 Top Programming Languages - IEEE Spectrum.” [Online]. Available: <http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages>. [Accessed: 15-Dec-2016].
- [23] “CSS | MDN.” [Online]. Available: <https://developer.mozilla.org/es/docs/Web/CSS>. [Accessed: 15-Dec-2016].
- [24] “Bootstrap · The world’s most popular mobile-first and responsive front-end framework.” [Online]. Available: <http://getbootstrap.com/>. [Accessed: 15-Dec-2016].
- [25] “Bootstrap 3 Tutorial.” [Online]. Available: <http://www.w3schools.com/bootstrap/>. [Accessed: 15-Dec-2016].
- [26] “Sublime Text: The text editor you’ll fall in love with.” [Online]. Available: <https://www.sublimetext.com/>. [Accessed: 15-Dec-2016].
- [27] “Emmet — the essential toolkit for web-developers.” [Online]. Available: <http://emmet.io/>. [Accessed: 15-Dec-2016].
- [28] “Brackets - A modern, open source code editor that understands web design.” [Online]. Available: <http://brackets.io/>. [Accessed: 15-Dec-2016].
- [29] “Atom - A hackable text editor for the 21st Century.” [Online]. Available: <https://atom.io/>. [Accessed: 15-Dec-2016].





- [30] “Eclipse - The Eclipse Foundation open source community website.” [Online]. Available: <https://eclipse.org/>. [Accessed: 15-Dec-2016].
- [31] “Mendeley | Free reference manager and academic social network | Elsevier.” [Online]. Available: <https://www.elsevier.com/solutions/mendeley>. [Accessed: 15-Dec-2016].
- [32] “Zotero | Home.” [Online]. Available: <https://www.zotero.org/>. [Accessed: 15-Dec-2016].
- [33] “Writer | LibreOffice en español - el paquete de oficina por excelencia.” [Online]. Available: <https://es.libreoffice.org/descubre/writer/>. [Accessed: 15-Dec-2016].
- [34] “Apache OpenOffice Writer.” [Online]. Available: <https://www.openoffice.org/es/producto/writer.html>. [Accessed: 15-Dec-2016].
- [35] “Microsoft Word | Software de procesamiento de texto y documentos.” [Online]. Available: <https://products.office.com/es-es/word>. [Accessed: 15-Dec-2016].
- [36] “LaTeX Editor—Home.” [Online]. Available: <http://www.latexeditor.org/>. [Accessed: 15-Dec-2016].

Capítulo V: Aspectos relevantes del desarrollo del proyecto

- [1] R. Axelrod, “Effective Choice in the Prisoner’s Dilemma,” *Source J. Confl. Resolut.*, vol. 24, no. 1, pp. 3–25, 1980.
- [2] “8.5. Understanding games people play: Axelrod’s tournament | Open Agent Based Modeling Consortium.” [Online]. Available: <https://www.openabm.org/book/3138/85-understanding-games-people-play-axelrods-tournament>. [Accessed: 22-Dec-2016].

Capítulo VI: Trabajos Relacionados

- [1] A. A. Deshpande and S. H. Huang, “Simulation Games in Engineering Education: A State-of-the-Art Review.”
- [2] J. M. Randel, B. A. Morris, C. D. Wetzel, and B. V. Whitehill, “The Effectiveness of Games for Educational Purposes: A Review of Recent Research.,” *Simul. Gaming*, vol. 23, no. 3, pp. 261–76, 1992.
- [3] P.-M. Léger, “Using a Simulation Game Approach to Teach Enterprise Resource Planning Concepts,” *J. Inf. Syst. Educ. Winter*, vol. 17, no. 4, 2006.
- [4] B. D. Mann, B. M. Eidelson, S. G. Fukuchi, S. A. Nissman, S. Robertson, and L. Jardines, “The development of an interactive game-based tool for learning surgical management algorithms via computer.”
- [5] S. G. Fukuchi, L. A. Offutt, J. Sacks, and B. D. Mann, “Teaching a multidisciplinary approach to cancer treatment during surgical clerkship via an interactive board game.”



- Am. J. Surg.*, vol. 179, no. 4, pp. 337–40, Apr. 2000.
- [6] P. L. Schumann, P. H. Anderson, and T. W. Scott, “Using Computer-Based Simulation Exercises to Teach Business Ethics,” *Teach. Bus. Ethics*, vol. 1, no. 2, pp. 163–181, 1997.
 - [7] S. W. Crown, “Improving Visualization Skills of Engineering Graphics Students Using Simple JavaScript Web Based Games,” *J. Eng. Educ.*, vol. 90, no. 3, pp. 347–355, Jul. 2001.
 - [8] B. Y. White, “Designing Computer Games to Help Physics Students Understand Newton’s Laws of Motion,” *Cogn. Instr.*, vol. 1, no. 1, pp. 69–108, Jan. 1984.
 - [9] E. Shifroni, D. Ginat, E. Shifroni, and D. Ginat, “Simulation game for teaching communications protocols,” *ACM SIGCSE Bull.*, vol. 29, no. 1, pp. 184–188, Mar. 1997.
 - [10] E. O. Navarro and A. Van Der Hoek, “SIMSE: AN INTERACTIVE SIMULATION GAME FOR SOFTWARE ENGINEERING EDUCATION.”
 - [11] E. Navarro and A. Van Der Hoek, “Multi-Site Evaluation of SimSE.”
 - [12] B. FRANK, “THE IMPACT OF CLASSROOM EXPERIMENTS ON THE LEARNING OF ECONOMICS: AN EMPIRICAL INVESTIGATION,” *Econ. Inq.*, vol. 35, no. 4, pp. 763–769, Oct. 1997.
 - [13] M. Pereda, D. Poza, J. I. Santos, and J. M. Galán, “Quality Uncertainty and Market Failure: An Interactive Model to Conduct Classroom Experiments,” Springer International Publishing, 2015, pp. 549–557.
 - [14] S. S. Izquierdo and L. R. Izquierdo, “The impact of quality uncertainty without asymmetric information on market efficiency.”

Capítulo VII: Conclusiones y líneas de trabajo futuras

- [1] R. M. Axelrod, *The complexity of cooperation : agent-based models of competition and collaboration*. Princeton University Press, 1997.





Impreso en Burgos el miércoles, 11 de enero de 2017